A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one partially covering the green one.

Conditional Statement and Looping

Alif Husnul Fikri

Conditional Statement

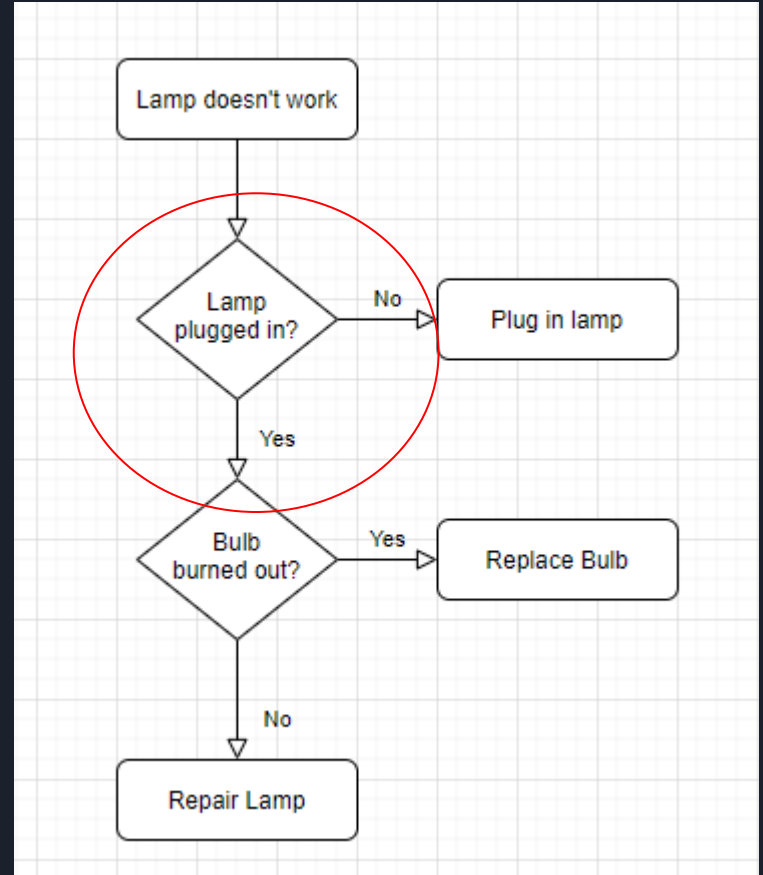
Merupakan syntax untuk menulis program dengan kondisi tertentu.

Conditional statement mensyaratkan kondisi yang hanya bisa dipenuhi atau tidak (binary, 0 atau 1, False atau True)

False atau True dalam python disebut Boolean

Syntax:

- If - else
- if - elif - else





Boolean

Merupakan tipe variable logika dengan value **True** atau **False**. Terdapat operasi boolean yang merupakan syntax untuk melakukan perbandingan value dalam suatu variable terhadap variable lainnya. Hasil yang akan dikeluarkan merupakan True or False

Operasi boolean:

Syntax	Penjelasan	Contoh
==	Sama dengan, A sama dengan B, A == B	A == B, 100 == 20, 100 == A
!=	Tidak sama dengan	A == B, 100 != 20, 100 == A
>, >=	Lebih besar, lebih besar sama dengan	A >= B, 100 > 20, 100 > B
<, <=	Kurang dari	A <= B, 100 < 20, 100 < B

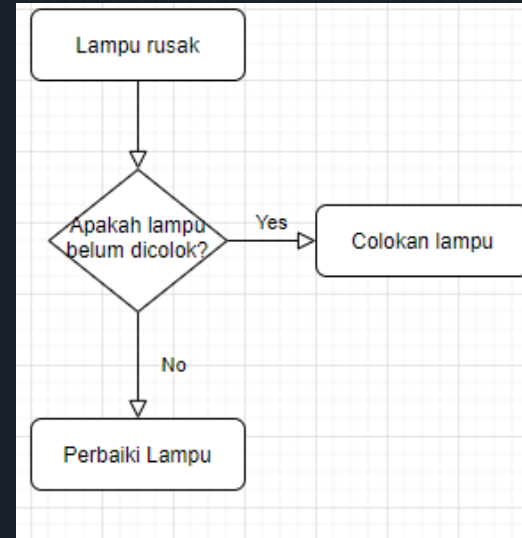
IF-ELSE Statement

Merupakan syntax untuk melakukan **conditional statement**.

```
if kondisi1:
    statement_1
    statement_2
else:
    statement_1
    statement_2
```

..... Disebut sebagai indent/indentasi, 1 indent -> 1 tab, 4 spasi

```
belum_dicolok = True
if belum_dicolok == True:
    print("silakan colokan lampu")
else:
    print("perbaiki lampu")
```



if -> Menandakan mulainya conditional statement

belum_dicolok == True -> merupakan kondisi yang harus dipenuhi

print("silakan colokan lampu") -> merupakan statement pertama dalam list

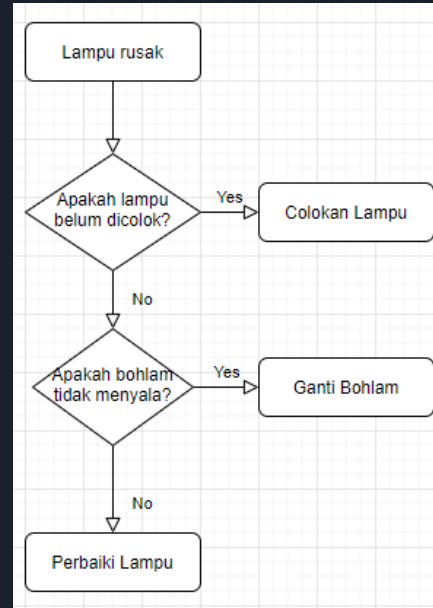
else -> Jika kondisi dipenuhi, maka lakukan statement di bawah

IF-ELIF-ELSE Statement

Merupakan syntax untuk melakukan **conditional statement**.

```
if kondisi1:
    statement_1
    Statement_2
elif kondisi2:
    statement_1
else:
    statement_1
```

```
belum_dicolok = True
bohlam_tidak_menyala = True
if belum_dicolok == True:
    print("silakan colokan lampu")
elif bohlam_tidak_menyala == True:
    print("silakan ganti bohlam")
else:
    print("perbaiki lampu")
```



if -> Menandakan mulainya conditional statement, jika kondisi pada if tidak terpenuhi, maka **program akan lanjut ke elif berikutnya**

elif -> **Else If**, merupakan conditional statement berikutnya jika kondisi pada if tidak terpenuhi. Jika kondisi terpenuhi, maka program akan **menjalankan statement** dalam **elif**.

IF-ELSE Statement

```
1  angka1 = input("Silakan Masukan Angka")
2  angka1 = int(angka1)
3  if angka1 <= 0:
4      print("Angka negatif")
5  elif angka1 == 0:
6      print("Angka Sama dengan 0")
7      print(angka1)
8  else:
9      print("Angka Positif")
```

IF-ELSE Statement

```
1  angka1 = input("Silakan Masukan Angka")
2  angka1 = int(angka1)
3  if angka1 <= 0:                If - kondisi 1
4      print("Angka negatif")    Statement jika kondisi1 terpenuhi
5  elif angka1 == 0:            elif - kondisi 2, jika kondisi 1 tidak terpenuhi
6      print("Angka Sama dengan 0") Statement jika kondisi1 terpenuhi
7      print(angka1)            Statement jika kondisi2 terpenuhi
8  else:                        elif - kondisi lain, jika kondisi 1 dan 2 tidak terpenuhi
9      print("Angka Positif")    Statement dalam else
```

Indentasi - menandakan statement yang berada di dalam if/elif/else

Statement akan dijalankan jika statement setelah ":" memiliki indentasi - 1 tab (4 spasi)

Program lain: ditandai dengan {}()



Condition

Kondisi dalam if dapat berupa berbagai macam hal. Selama kondisi if terpenuhi, maka statement akan dijalankan.

If True:	Kondisi terpenuhi
flag = True If flag == True:	Karena flag bernilai True, kondisi terpenuhi
flag = True If flag:	Karena flag bernilai True, maka seperti if True
If angka1 < 10:	Akan berjalan jika angka1 < 10. Jika Angka1 = 0, maka angka1 < 10 akan sama dengan True



AND, OR, IN, NOT

kondisi1 **and** kondisi2 -> jika **kedua** statement terpenuhi, maka akan menghasilkan True

kondisi1 **or** kondisi2 -> jika **salah satu** statement terpenuhi, maka akan menghasilkan True

var_value **in** var_list -> Jika angka/huruf/value dalam variable **var_value** ada di dalam variable **var_list**, maka akan menghasilkan true

not kondisi1 -> jika kondisi1 tidak terpenuhi, maka akan menghasilkan True



A & B (dua2nya terpenuhi, hasilnya True)

A, B, A dan B

True, True, True

True, False, False

False, True, False

False, False, False

A atau B (salah satu terpenuhi, hasilnya True)

A, B, A atau B

True, True, True

True, False, True

False, True, True

False, False, False



Exercise

Dengan menggunakan input + if else/if elif else

Program konversi Temperature

- User diminta input temperature dalam C, dan unit konversi
- Jika unit konversi sm dengan R, hasil = $\frac{4}{5} C$
- Jika unit konversi sm dengan F, hasil = $\frac{9}{5} C + 32$
- Jika unit konversi sm dengan K, hasil = $C + 273.15$
- Print hasil

Program pengecekan input

- User diminta untuk input jam terbang
- Jika jam tidak sesuai, maka print "input salah"



Exercise

Dengan menggunakan input + if else/if elif else

Dengan menggunakan mod (%)

1. Print apa yang terjadi jika $1 \% 4$, $12 \% 2$, $11 \% 2$
2. User diminta input angka
3. Jika angka genap = print "angka genap"
4. Jika angka ganjil = print "angka ganjil"

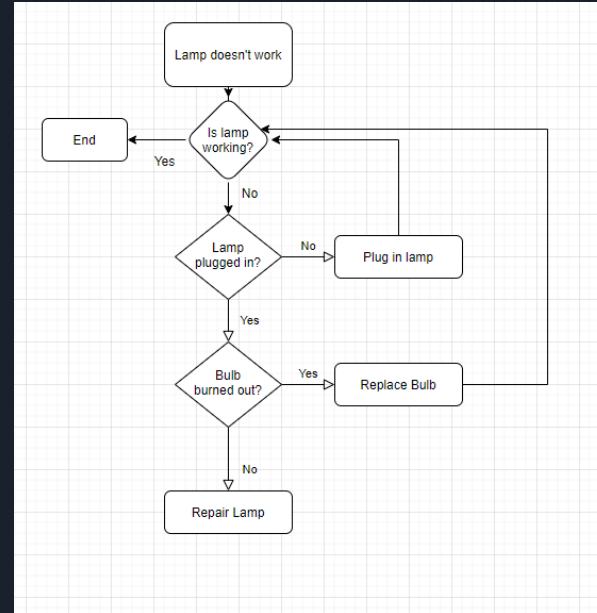
1. User diminta input angka/tahun
2. Tahun tidak boleh < 2010
3. Jika input tahun yg diberikan user merupakan tahun kabisat (habis dibagi 4), maka print "tahun kabisat", jika bukan print "tahun basit"

Looping

Proses yang terus mengulang hingga mencapai suatu kondisi.

Ada dua tipe looping yaitu:

- While
- For





While Loop

While loop adalah proses pengulangan yang terus dilakukan selama **kondisi terpenuhi**

```
while kondisi1:  
    statement_1  
    Statement_2  
else:  
    statement_1
```

```
while angka < 10:  
    print(angka)  
    angka = angka +  
1  
else:  
    print("berhenti")  
)
```

Dalam proses looping, terdapat infinite looping. Infinite looping adalah looping yang tidak akan berhenti. Program yang baik tidak akan menciptakan infinite looping!



For Loop

For loop adalah pengulangan yang dilakukan untuk **sebuah urutan** (akhir value dalam list angka, akhir baris dalam sebuah data, akhir index/elemen dalam list)

```
for nama_var in urutan:  
    statement_1  
    statement_2
```

```
list_a = [1,2,3,4]  
for value in list_a:  
    print(value)  
    print(value+2)
```

For -> memulai looping

nama_var-> dummy variable, dapat diganti dengan nama lainnya, hanya untuk digunakan sebagai panggilan untuk value dalam list

urutan -> nama **iterable** variable (list/set/string/dict)



Looping + IF

If dan looping dapat digabungkan menjadi satu, sebagai contoh

```
list_angka = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for value in list_angka:
    print(value)
    if value > 5:
        print("value lebih dari 5")
    else:
        print("value kurang dari sama dengan 5")
print("-- berikutnya --")
```


Looping + IF

If dan looping dapat digabungkan menjadi satu, sebagai contoh

```
list_angka = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for value in list_angka:
    print(value)
    if value > 5:
        print("value lebih dari 5")
    else:
        print("value kurang dari sama dengan 5")
print("-- berikutnya --")
```

Looping value dalam list. Nilai dalam list angka akan dipanggil sebagai "value"

Statement dalam if, yang merupakan bagian dalam for

Statement dalam for, berada di luar if

Looping For - Variasi

```
8 for angka in range(0, 5, 1):
9     print(angka)
10
11 dict_a = {'nomor': [1,2,3,4,5],
12           'nama': ['alif', 'laura', 'andi', 'nanda', 'radit']}
13
14 for val in dict_a.values():
15     print(val)
16
17 for key in dict_a.keys():
18     print(key)
19
20 for key, val in dict_a.items():
21     print(key, val)
22
23 list_angka = list(range(1, 100, 10))
24 for index, angka in enumerate(list_angka):
25     print(index, angka)
26
```

Looping menggunakan range

Looping dict

- Value dalam dict
- Keys dalam dict
- Key dan Value dalam dict

Enumerate: looping dengan mengeluarkan index dan value dari sebuah *iterable*.

Break - Continue

Looping dapat dihentikan dengan menggunakan **break**.

Looping dapat **dipaksa** lanjut dengan menggunakan **continue**

```
6
7 data_buah = ['pisang', 'apel', 'melon', 'semangka']
8 for buah in data_buah:
9     print(buah)
10    if buah == 'melon':
11        print('melon ditemukan')
12        break
```

Jika melon ditemukan,
maka program akan
berhenti setelah print
"melon ditemukan"

```
6
7 data_angka = [1,2,3,4,5]
8 for angka in data_angka:
9     print("angka", angka)
10    if angka == 3:
11        continue
12    print("angka ditambah angka: ", angka + angka)
13
```

Jika angka 3 ditemukan, maka
looping dipaksa lanjut ke value
berikutnya dalam list (statement 12
tidak dijalankan)



Break - cara lain

Cara lain membuat break adalah dengan menggunakan variable sebagai **flag** (penanda)

```
7 n = 0
8 break_flag = True
9 while break_flag:
10     print(n)
11     if n == 10:
12         break_flag = False
13     n = n + 1
```